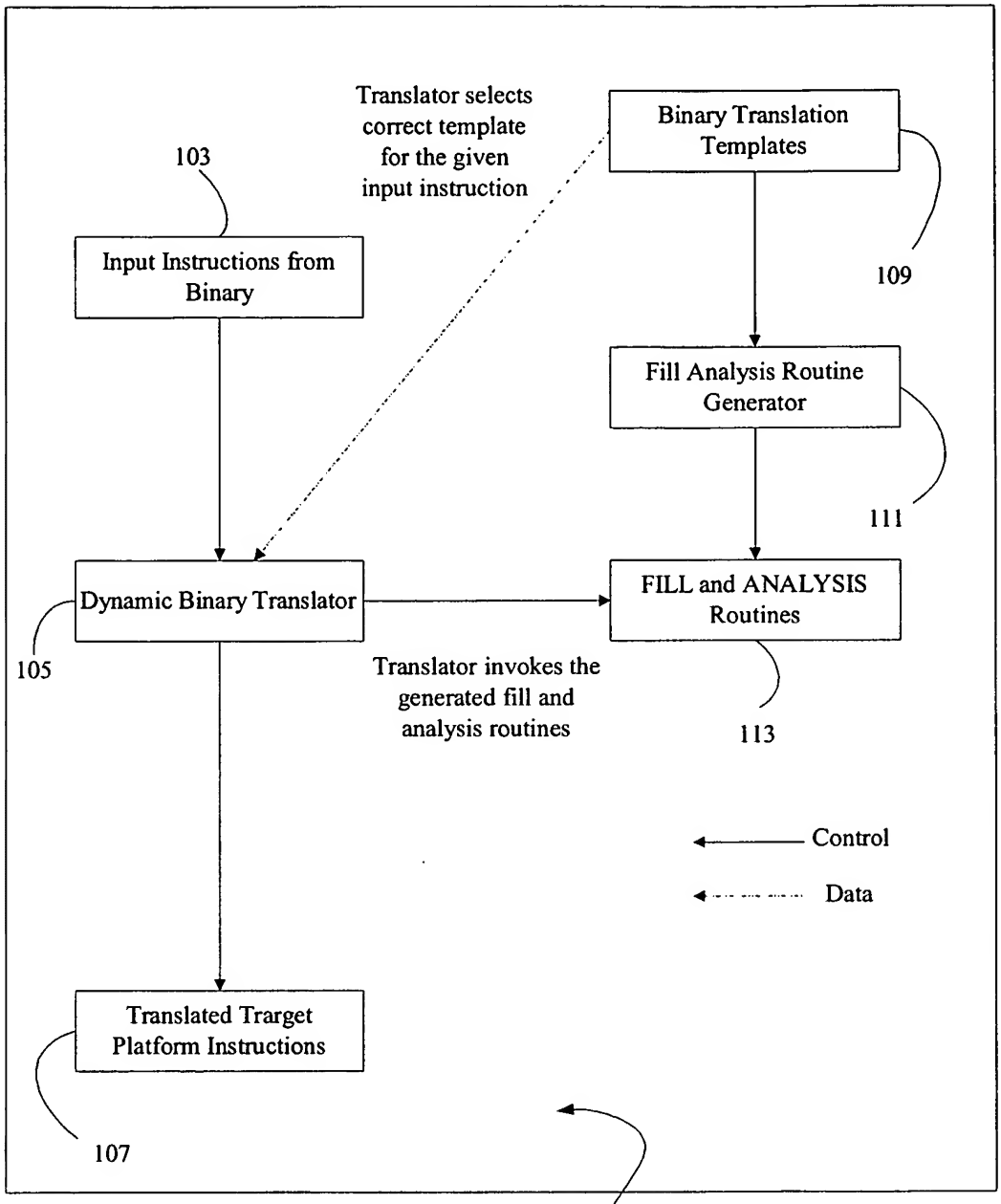
Translator selects correct template for the given input instruction
Binary Translation Templates
103
Input Instructions from Binary
109
Fill Analysis Routine Generator
111
Dynamic Binary Translator
FILL and ANALYSIS Routines
105
Translator invokes the generated fill and analysis routines
113
Control
Data
Translated Trarget Platform Instructions
107
101

Figure 1

	Opcode	SrcReg2	SrcReg1	Condition	Direction	subpeode				Width	Target
ADD	0x02	r2	r1	c	f	e1	1	e2	0	d	t

Figure 2

	Opcode	Condition	Direction	subopeode	Width
ADD R1,R2,R5	0x02	0	0	0x18	0
ADD, <> *R1,R2, R5	0x02	1	1	0x18	1
ADD, C, R1, R2, R5	0x02	0	0	0x1c	0
ADD, L*R1, R2, R5	0x02	0	0	0x28	1

Figure 3

```
.proc template _ADD_L_EQ
magic_04540              ::
Template_ADD_L_EQ        ::
{
.mii
                         (p0) add    P.A_t   =  PA_r1  ,PA_r2  ;;  // INSTR1
                         (p0)        PT,pF   =  r0     ,PA_t      // INSTR2
                         cmp4.eq
                         Nop.i                                ;;  // INSTR3
}
.endp
template_ADD_L_EQ
```

Figure 4

```
// To deposit bits 20:25 from a 32 bit Pa -RISC instruction "pa_instru", into bits 10:15 in 41 bit
// Itanium instruction "ia_instr":

((struct {u64 padl:48;u64 f:6; u64 pad 2:10} *) &la_instr)  ->f =
                                    ((struct {u32 padl: 13;u32 f:6; u32 pad 2:21} *) &  pa_instr)->f
```

Figure 5

```
//   Instr 1:    (p0)add          PA_t      =   PA_rl      ,   PA_r2
//   Instru 2:   (p0)emp4.eq      p7,p6     =   r0         ,   PA_t


1.    Add WAW edge between nodes (last_producer[PA -t], Instr 1)
2.    Add RAW edge between nodes (last_producer[PA_rl], Instr 1)
3.    Add RAW edge between nodes (last_producer[PA_r2],  Instr 1)
4.    Add RAW edge between notes (last_producer[qualifying -predicate], Instr 1)
5.    Update Instr 1 as last_producer [PA_t]
6.    Update Instr 1 as live_user [PA_t], live_user[PA_rl] and live_user [PA_r2]
7.    Update miscellaneous information about Instr 1 (viz. Instr uction type, issue type,
      latency info etc). for use by schedular at runtime.

//    using the static invartant that there is a RAW dependency
//    between Instr 1, Instr 2 on PA_1

8.    add Raw edge between nodes (Instr 1, Instr 2)
9.    Add RAW edge between nodes (last -producer[qualifying-predicate], Inst 2)
10.   Add WAW edge between nodes (last -producer[p6, Instr 2)
11.   Add WAW edge between nodes (last -producer[p7], Instr 2)
12.   Update Instr 2 as last_producer[p6], and last_producer [p7]
13.   Update Instr 2 as live_user [PA_t]
14.   Update miscellaneous information about Instr 2 (viz. Instruction type, issue type,
      latency info etc.) for use by scheduler at runtime.
```

Figure 6

Name	PA-RISC Instructions Translated	Without Templates		With Templates		Translation Speedup
		CPU Cycles Overhead (in million)	CPU Cycles per PA Instruction	CUP Cycles Overhead (in million)	CPU Cycles per PA Instruction	
Netscape	1139954	1208	1060	232	204	25.19
Acroread	4427072	2190	579	549	126	4.55
Java-Version	21600959	314	145	571	26	5.50
_200_check	36295360	2855	78	547	14	5.27
_201_Compress	59032516	57197	968	12715	215	4.49
Gzip	132250	125	949	22	172	5.51
Tar	160691	176	925	33	176	5.23
Ghostview	534737	690	1291	120	224	5.74
Gcc	1343130	2085	1553	571	425	3.65
Parser	148075	251	1701	54	368	4.61

Figure 7

Application	Time is Secs w/o Templates	Tim in Secs with Templates	Overall Speedup Factor
Netscape	2.57	0.58	4.37
Acrobat	17.33	5.90	2.80
Java – version	34.29	5.89	5.82
_200_check	10.224	3.903	2.61
_202_jess	380.241	126.833	2.99
_209_db	1040.341	145.379	7.15
_213_javac	2602.306	822.393	3.16
_227_mtrt	452.022	111.609	4.11
Gzip	2.57	4.11	0.62
Tar	1.26	1.04	1.24
Ghostview	5.64	3.2	1.76
Gcc	474	614	0.77
Parser	1217	1228	1.00

Figure 8